Faster Multi-Object Segmentation using Parallel Quadratic Pseudo-Boolean Optimization

Niels Jeppesen, Patrick M. Jensen, Anders N. Christensen, Anders B. Dahl, and Vedrana A. Dahl

Visual Computing, DTU Compute. Technical University of Denmark, Kongens Lyngby, Denmark - {niejep,patmjen,anym,abda,vand}@dtu.dk

Summary

In this paper we:

- Introduce a parallel algorithm for solving quadratic pseudo-Boolean optimization (QPBO) problems.
- Demonstrate an 11x and 19x speed-up for large problems and ~3x for small problems.
- Demonstrate our method outperforms current state of the art algorithms by a large margin.

We focus on image segmentation, but our algorithm can solve any QPBO problem. Our implementation is available at: github.com/Skielex/shrdr

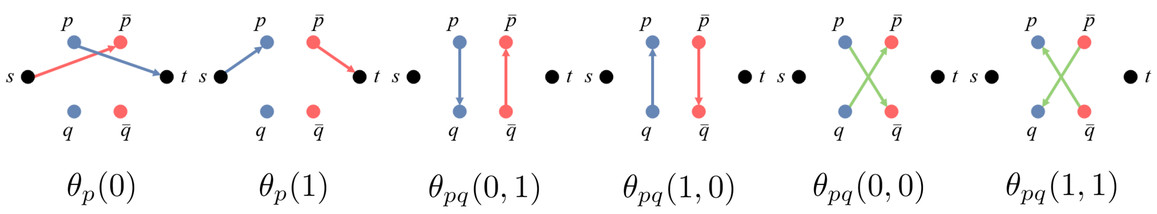
Method

Graph Construction

Given a quadratic pseudo-Boolean energy function

$$E(\mathbf{x}) = \sum_{p \in \mathcal{V}} \theta_p(x_p) + \sum_{p,q \in \mathcal{V}} \theta_{pq}(x_p, x_q)$$

a graph is constructed as follows



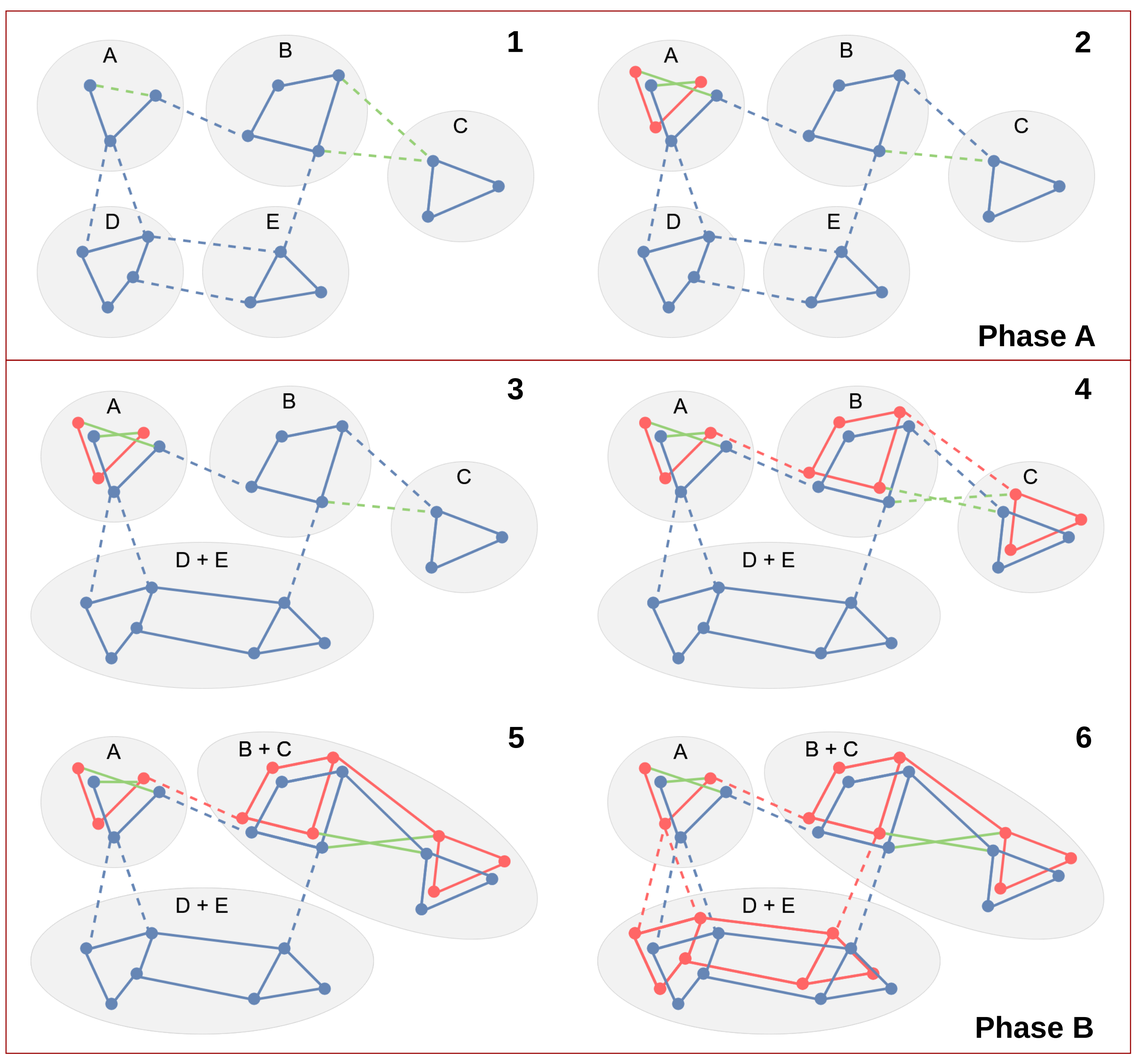
Min-cut/max-flow of this graph gives the minimizer of the energy function.

Parallel QPBO Algorithm

We combine the original QPBO algorithm with the block-based parallelization approach by Liu and Sun. The algorithm has two phases:

A. Split the problem into disjoint blocks and solve each block in parallel.
B. Merge pairs of blocks and re-solve. Repeat until all blocks are merged.

During both phases, graph arcs for non-submodular terms are only added when needed. This is key to our algorithm's performance.



Speed-up Results

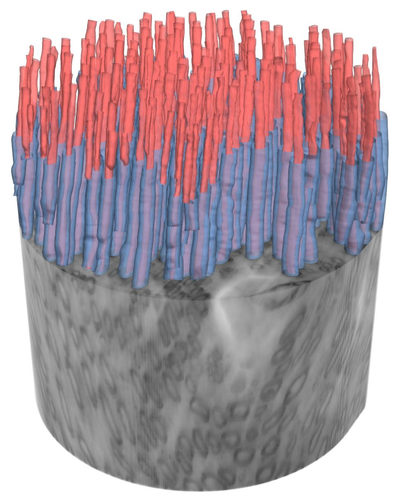
Data & Setup

We apply our algorithm to two different tasks:

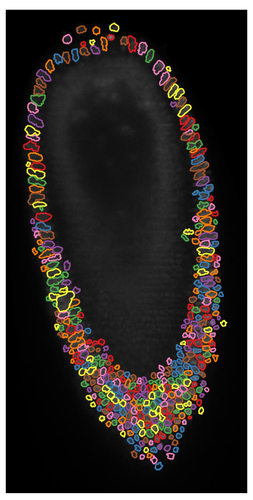
1. Nerves (N1, N2): segment nerves from a large 3D µCT image.

2. Broad Bioimage Benchmark (BBB): segment cells from many small 2D images.

All tests are run on a dual socket machine with two 16 core Intel Xeon 6226R CPUs and enough RAM to keep all data loaded in memory.

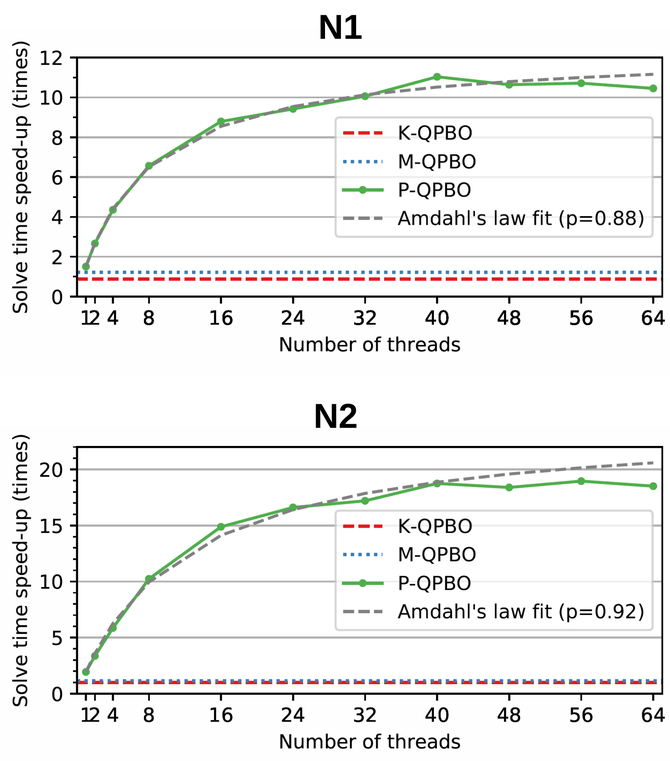


Nerves



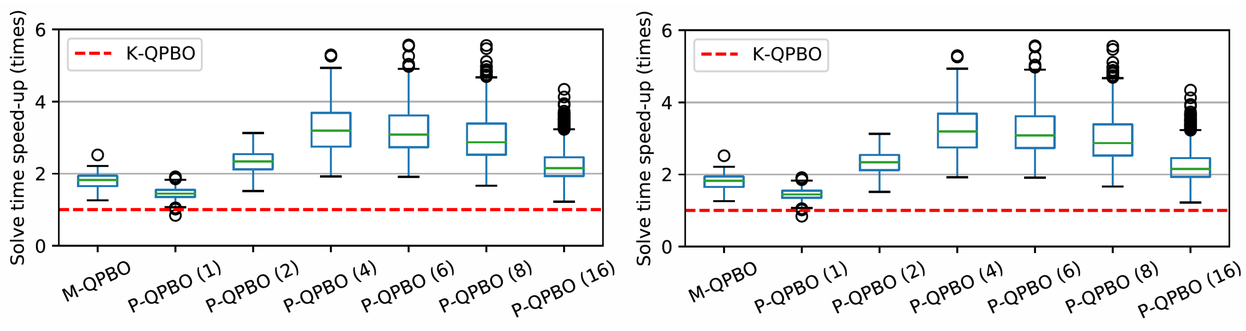
Broad Bioimage Benchmark

Results on Nerves (Large Problem)



	N1	N2
Nodes	364 M	818 M
Edges	2,124 M	4,864 M
Memory footprint		
K-QPBO	134.2 GB	306.8 GB
M-QPBO	60.1 GB	182.7 GB
P-QPBO	70.0 GB	224.9 GB
Fastest solve time		
K-QPBO	844 s	4,534 s
M-QPBO	638 s	3,897 s
P-QPBO (1)	561 s	2,338 s
P-QPBO (16)	96 s	305 s
P-QPBO (32)	83 s	264 s
P-QPBO (40)	**76 s**	242 s
P-QPBO (56)	79 s	**239 s**

Results on Broad Bioimage Benchmark (Small Problems)



Performance Comparison

We compare our algorithm with:

- **Liu-Sun:** parallel min-cut/max-flow solver.
- **EIBFS:** state of the art serial min-cut/max-flow solver.
- **P-EIBFS:** parallelized version of EIBFS.
- **EIBFS-QPBO:** estimate of QPBO with EIBFS.

***Tested with up to 32 threads.**

	N1 Mem.	Best speed-up	
		N1	BBB
M-QPBO	60.1 GB	1.32	1.72±0.28
P-QPBO (32)	70.0 GB	**10.17**	**2.96±0.89**
Liu-Sun (32)	70.0 GB	4.42	0.78±0.13
EIBFS	175.1 GB	0.92	0.33±0.08
P-EIBFS (32)	175.8 GB	0.59	0.68±0.14
EIBFS-QPBO	175.1 GB	1.83	0.66±0.08